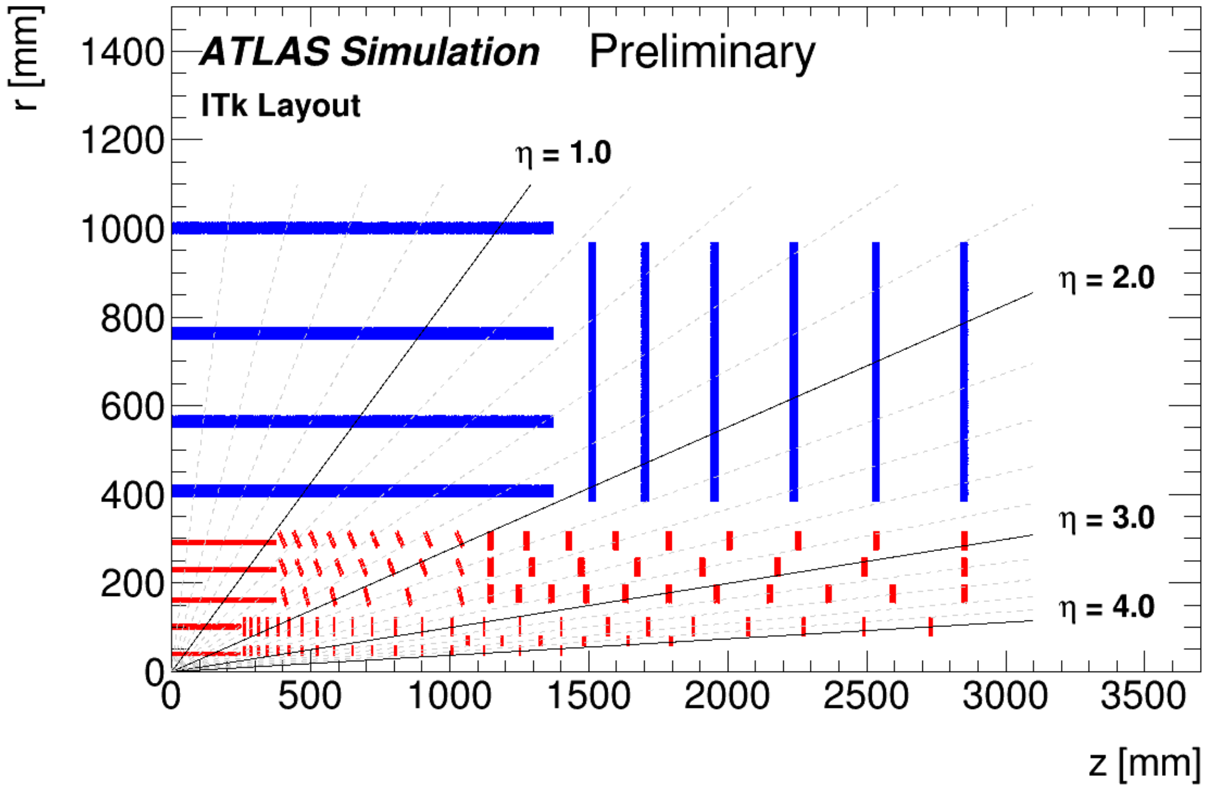
TPU for Exa-TrkX

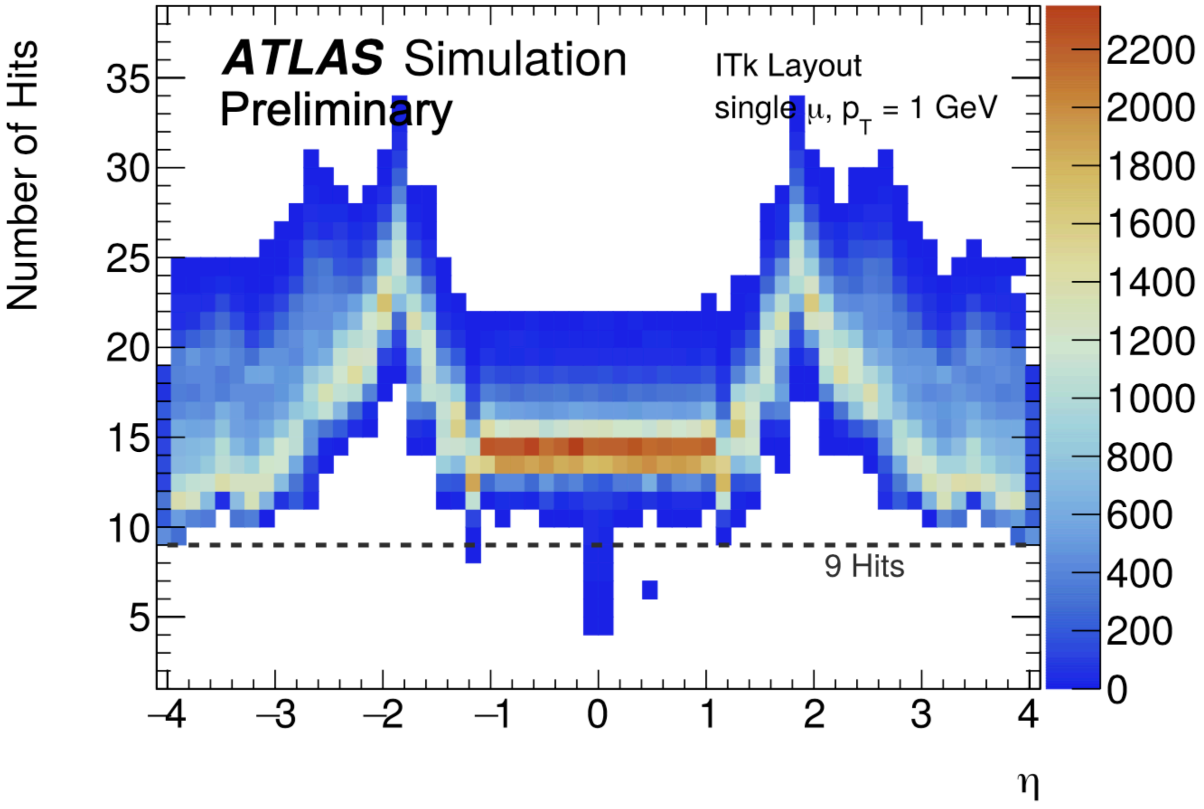
Xiangyang Ju

ExaTrkX Collaboration Meeting
7 April 2020

Introduction

- HL-Luminosity LHC starts operations in ~2027, to reach a peak instantaneous luminosity of 7 × 10^{34} cm^{-2} s^{-1}, corresponding to ~200 proton-proton collisions per bunch crossing
- Each collision produces about 10,000 particles





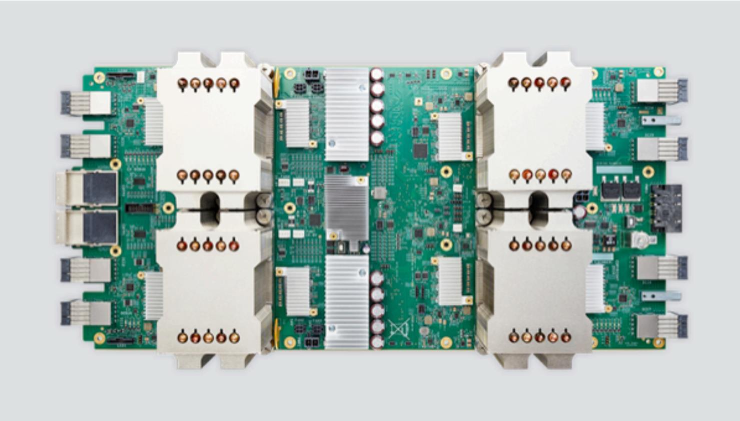
- The ATLK Inner Tracker will record ~150,000 hits for each event.
- For doublet graph, 150,000 nodes and 135,000 true edges. Assuming the fake rate of input doublets is 10%, the doublet graph would have 150,000 nodes and 1,350,000 edges.

Tensor Processing Units

- Why not GPUs?
  - Limit amount of high bandwidth memory (HBM). NVIDIA V100 GPU has 32 GB HBM
  - Need to split the whole graph into small segments and feed each segment to GPU
- Why TPUs?
  - primarily because of its large HBM, which can reach 32 TB
  - specially designed for the matrix operations, particularly the matrix multiplications, which happens a lot in the bit graph
  - one can run TensorFlow and Pytorch (via pytorch/xla)
  - drawbacks:
    - does not support all TensorFlow operations
    - does not support double-precision arithmetic

Cloud TPU offering

Colab and Kaggle provides limited but free access to TPU, good places for debugging.

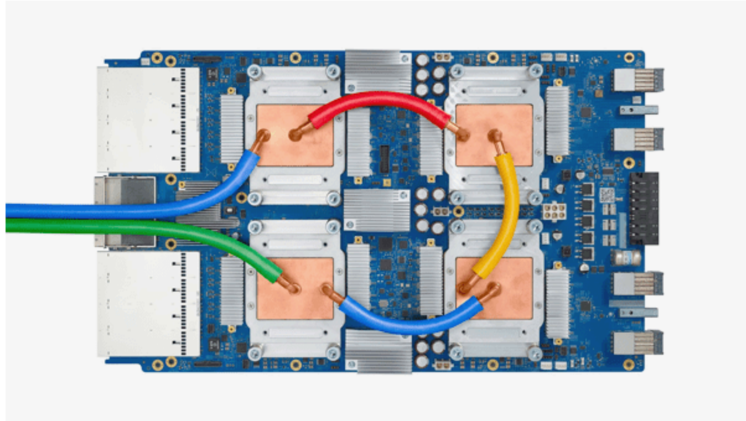


Cloud TPU v2

180 teraflops

64 GB High Bandwidth Memory (HBM)

$4.5/hour

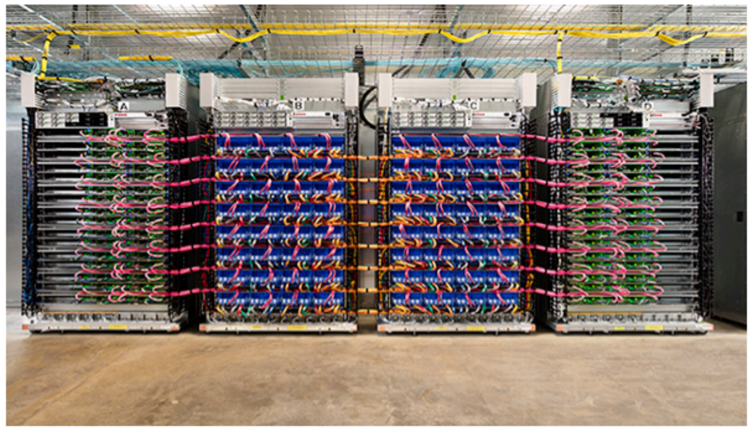


Cloud TPU v3

420 teraflops

128 GB HBM

$8.0/hour



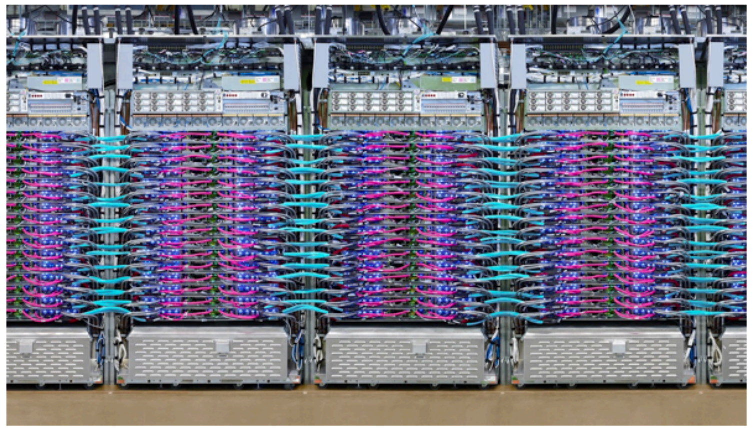
Cloud TPU v2 Pod

11.5 petaflops

4 TB HBM

2-D toroidal mesh network

$384/hour



Cloud TPU v3 Pod

100+ petaflops

32 TB HBM

2-D toroidal mesh network

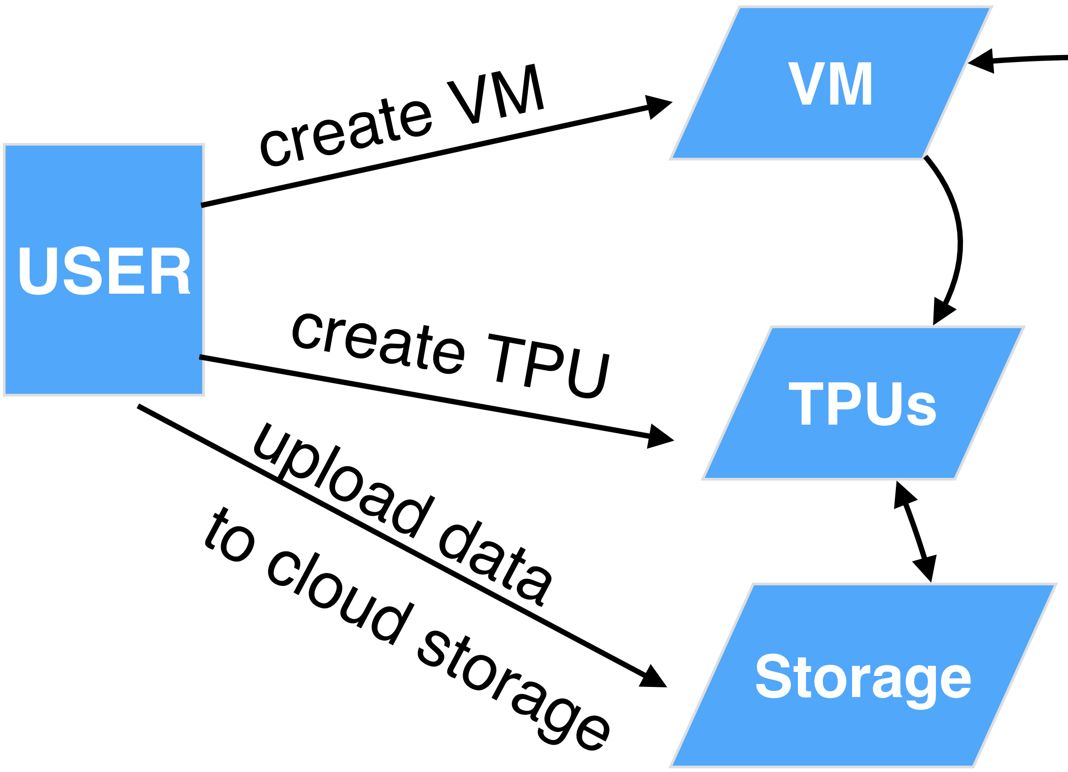
contact sales

Migrating to cloud TPU

To reach best performance, TPU prefers

- batch size that are multiples of 8, because a single could TPU consists of 8 TPU cores
- fixed shapes, so dynamic graphs are not supported
  - padding graph is added for each doublet graph so that the number of nodes and edges are constant values
- matrix dimension of 128, because the structure of the matrix unit hardware is a 128x128 systolic array
  - Systolic array: hard-wired processing units for specific operations
- training data in the cloud at the same zone
  - before training, upload the data to google cloud storage that sits in the same zone as the cloud TPU

Using cloud TPU



- Install python packages and scripts,
- In the training code:
  - create the TPUStrategy to use the TPUs
  - point TFRecord the cloud storage directory for training inputs
  - perform the training

- Just made the GNN model run on TPU with some caveats to resolve
  - remove the padding graph from the loss calculations
  - find a workaround to replace the weighted log_loss
- Next step is to figure out which TPU type we need so that we could use one graph for one event in the training